Docket No.: M4065.0910/P910

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR U.S. LETTERS PATENT

Title:

TESTING CMOS TERNARY CAM WITH REDUNDANCY

Inventor

Prasad Mantri

DICKSTEIN SHAPIRO MORIN &
OSHINSKY LLP
2101 L Street NW
Washington, DC 20037-1526
(202) 828-2232

TESTING CMOS TERNARY CAM WITH REDUNDANCY

FIELD OF INVENTION

[0001] The present invention relates generally to semiconductor memory devices and, more particularly to a method for testing CMOS Ternary Content Addressable Memory (TCAM) devices. The present invention may also be used to test binary content addressable memory (CAM) devices.

BACKGROUND OF THE INVENTION

[0002] An essential semiconductor device is semiconductor memory, such as a random access memory (RAM) device. A RAM allows a memory circuit to execute both read and write operations on its memory cells. Typical examples of RAM devices include dynamic random access memory (DRAM) and static random access memory (SRAM).

[0003] Another form of memory is the content addressable memory (CAM) device. A CAM is a memory device that accelerates any application requiring fast searches of a database, list, or pattern, such as in database machines, image or voice recognition, or computer and communication networks. CAMs provide benefits over other memory search algorithms by simultaneously comparing the desired information (i.e., data in the comparand register) against the entire list of pre-stored entries. As a result of their unique searching algorithm, CAM devices are frequently employed in network equipment, particularly routers and switches, computer systems and other devices that require rapid content searching.

[0004] In order to perform a memory search in the above-identified manner, CAMs are organized differently than other memory devices (e.g., DRAM). For example, data is stored in a RAM in a particular location, called an address. During a memory access, the user supplies an address and writes into or reads the data at the specified address.

[0005] In a CAM, however, data is stored in locations in a somewhat random fashion. The locations can be selected by an address bus, or the data can be written into the first empty memory location. These data written into the memory location are called as "rules". Every memory location includes one or more status bits which maintain state information regarding the memory location. For example, each memory location may include a valid bit whose state indicate whether the memory location stores valid information, or whether the memory location does not contain valid information (and is therefore available for writing).

[0006] In a ternary CAM every bit of data has a Data and Mask bit. Data for a bit is the data that has to be compared for that bit, this data is however qualified by a mask bit which is used as a control bit for each of the data bit to enable or disable it from comparison. When the bit is masked it is called don't care bit. Following table shows a one possible data and mask scheme

Name	Data	Mask
No Mask "0"	0	1
No Mask "1"	1	1
Mask "0"	0	0
Mask "1"	1	0

Once information is stored in a memory location, it is found by comparing every bit in a memory location with corresponding bits in a comparand register. When the content stored in the CAM memory location does not match the data in the comparand register, a local match detection circuit returns a no match indication. When the content stored in the CAM memory location matches the data in the comparand register, the local match detection circuit returns a match indication. If one or more local match detect circuits return a match indication, the CAM device returns a "match" indication. Otherwise, the CAM device returns a "no-match" indication. In addition, the CAM may return the

identification of the address location in which desired data is stored or identification of one of such addresses if more than one address contained matching data. Thus, with a CAM, the user supplies the data and gets back an address if there is a match found in memory. In a ternary CAM when data with mask are stored at various locations and searches are done with a data then for a given data, there is a possibility of multiple rules getting matched for a given key. In a Ternary CAM the process of resolving multiple matches to generate a single match output is called priority resolution. There are many schemes of priority resolution two of the ones are highest order priority or lowest order priority. Highest Priority resolution is where the largest address matched is given out and the lowest order priority is where the smallest address match is given out.

[0007] Fig. 1 is a circuit diagram illustrating a conventional SRAM based ternary CAM cell 100. The CAM cell 100 includes a data storage portion 110, which is comprised of an access transistor D0 and a pair of inverters D1a, D1b, arranged in as shown in Fig. 1. The gate of the access transistor D0 is coupled to a word line WL, and one source/drain terminal of the access transistor D0 is coupled to a bit line BL. Thus, a voltage level indicative of a logical value that is placed on the bit line BL is gated across the access transistor D0 when the word line WL is high. This sets the voltage at node D to the voltage equal to the logical value. The two inverter structure D1a, D1b subsequently maintains the logical value at node D and a complement of the logical value at node D#.

[0008] A match section 120 of the CAM cell 100 is comprised of transistors M1, M1#, M2, M2# and M3, which controllably couple the match line ML to ground in certain situations. In the CAM cell 100, a "no-match" condition is detected if the match line ML is pulled to ground during the match operation, while a "match" condition is detected if the match line ML remains at a precharge potential. Typically, the detection of the state of the match line ML is performed by a sense amplifier (not illustrated).

[0009] Transistor M3 has one source/drain terminal coupled to the mach line ML and another source/drain terminal coupled to a four-transistor circuit formed by transistors

M1, M1#, M2, and M2#. The gate of transistor M3 is coupled to a register M, which stores a mask value. The register M and transistor M3 are present when CAM cell 100 is a ternary CAM (TCAM) cell. TCAM cells permit the search expression to include "don't care" bits, while binary CAMs enforce a match/no-match evaluation of every bit. If CAM cell 100 were a binary CAM, transistor M3 and register M would not be present and the source/drain terminals of transistor M2 and M2# illustrated as being coupled to a source/drain terminal of transistor M3 would instead be coupled directly to the match line ML.

[0010] Transistors M1, M1#, M2, and M2# form a comparison circuit. The gates of transistors M2, M2# are respectively coupled to search data lines SD and SD#. The search data is placed on the line SD while the complement of the search data is placed on the line SD#. Similarly, the logical value stored at node D is coupled to the gate of transistor M1 while the complement of that logical value stored at node D# is coupled to the gate of transistor M1#. In this manner, the comparison structure will pull the match line ML voltage from its precharged level to ground if and only if the search data on line SD matches the data stored at node D and the complement of the search data on line SD# matches the complement of the data stored at node D, assuming that transistor M3 is conducting. If transistor M3 is not conducting, the match line ML potential will remain at its pre-charged level to force a "match" condition.

[0011] When a memory device is manufactured, it must be tested. Although testing is time consuming and costly, testing is required to identify errors in the device. If errors are not identified, the use of the memory device can corrupt data. In addition to error detection, an ideal test should also be able to inform the tester as to which portion of a CAM cell is defective. If many defects are isolated to a same problem area, the data can form the basis of improving the device fabrication process. Accordingly, there is a need for a method to efficiently test CAM cells in a CAM device and to identify which component of a defective CAM cell failed.

SUMMARY OF THE INVENTION

[0012] The invention is directed to a method for testing a CMOS ternary content addressable memory (TCAM) device that has redundant CAM rows and columns. The method includes a match line test to identify stuck match lines, followed by a walking "1" pattern across the columns to identify weak pull downs (from the match line to ground), and is followed by a row-by-row match test. If stuck matched lines are identified then those lines are masked preventing them from participating in priority encoding and the defective row is replaced by a redundant CAM row, in case redundant resources are available. During the row-by-row match test, a failed CAM cell can be repaired or the row associated with the failed CAM cell can be disabled. Additionally, individual CAM cells that are identified as being defective can be further tested to identify which component of the CAM cell failed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The foregoing and other advantages and features of the invention will become more apparent from the detailed description of exemplary embodiments of the invention given below with reference to the accompanying drawings, in which:

[0014] Fig. 1 is a circuit diagram of a conventional CAM cell;

[0015] Fig. 2 is a block diagram of a CAM device incorporating the CAM cell of Fig. 1;

[0016] Fig. 3 is a flowchart illustrating the testing method of an exemplary embodiment of the invention;

[0017] Fig. 4 is a flowchart illustrating a test for identifying stuck match lines according to an embodiment of the invention;

[0018] Fig. 5 is a flowchart illustrating a test for identifying weak pull downs between the match line and ground according to an embodiment of the invention; and

[0019] Figs. 6A and 6B are flowcharts illustrating the row-by-row match test according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0020] Now referring to the drawings, where like reference numerals designate like elements, there is shown in Fig. 2 a CAM device 210 and a tester 250. The CAM device 210 includes at least one memory array 220. Each array 220 has a plurality of CAM cells 100 (Fig. 1). The CAM device 210 also includes a control circuit 230 for operating the CAM device 210 and for interfacing with external devices. The tester 250 includes an interface 251 for communicating with the CAM device 210. The tester 250 further includes a processor 252, which can execute a series of tests on the CAM device 210 as described below with reference to Figs. 3-5, 6A, and 6B. The tester 250 tests the device 210 by placing the device 210 into a test mode and then executing the below described procedure. The device 210 may be placed into test mode in a variety of ways which are well known, such as asserting a signal on a test mode enable pin (such pins are typically made non accessible after the device 210 has been tested) or placing the device 210 into the test mode using an instruction. The invention is not be limited by the manner in which the device 210 is placed into test mode.

[0021] Fig. 3 is a high level flowchart illustrating the test method of an embodiment of the invention. The tester 250 executes each of the below described steps by sending an appropriate signal to the CAM device 210. At step S31, the CAM device 210 is tested for

stuck match lines (described in more detail below). Execution then continues at step S32, where weak pull-down lines are identified (described in more detail below). At step S33, a row-by-row test is executed to ensure each location in the CAM device 210 can be written and read correctly (described in more detail below). The stuck match line test and weak pull-down line tests are executed first in order to identify faulty match lines and pull-down lines, which if not accounted for at the beginning of the testing procedure can cause later tests to return false results. Finally, at step S34, each defective cell identified can be further analyzed under a fault model to determine which component of the cell failed.

[0022] In the following discussion we use the term 1-pattern and 0-pattern and these patterns are described as follows. The patterns can be of various forms depending upon the physical organization of the CAM cells and the coupling faults that are being exercised. The simplest 1-pattern will be of the form (111111.....) when there is no consideration of the coupling faults or for physical organization. When the logical to physical organization is the same then a pattern applied is a checkered board pattern where the alternate bits are compliment of each other, the pattern is of the form (01010101.....). When there is column muxing the logical checkered board is so arranged so that we have physically a pattern where the a bit is physically surrounded by its compliment. A 0-pattern is the bit wise compliment of a 1-pattern.

[0023] Fig. 4 is a flowchart illustrating an exemplary stuck match line test S31 according to an embodiment of the invention. The objective of this test is to identify and mask for any stuck match lines to remove the stuck match lines from the CAM and replace them with a redundant match row. The test S31 begins at step S41, where the data storage location 110 of each CAM cell 100 is written with a 1-pattern. Execution continues at step S42, where the search mask M of each to not mask state (to consider every cell to participate in match). (If device 210 is a binary CAM, there is no mask bit, so step S42 would not be performed. Rather, execution would continue at step S43 after S41.) This ensures that every CAM cell 100 will participate in the search by enabling the comparison portion 120 of each CAM cell 100 to pull down its associated match line in case of a

mismatch. At step S43, the CAM cell 100 is searched using a 0-pattern search data pattern. At step S44, an inquiry is made to see if there were any matches. In a correctly operating device 210, there should be no matches since step S41 set every data location to a logical one and the search data contained zeros. Thus, if there are no matches, the test completes. A match is an indication of a faulty match line. If there are multiple faulty match lines then the highest priority matched line will be the match address. The address line associated with the matching entry is therefore disabled (S45). If there is a redundant match line, it can be remapped to replace of the disabled match line. After step S45, execution resumes at step S43. Thus, steps S43, S44, and S45 are repeatedly performed in a loop until the test S31 ends with no matches or until all redundant resources are exhausted, when the CAM is declared as unrepairable. This process can be repeated with the CAM filled with 0-pattern and a 1-pattern as the match pattern. This pattern though not strictly necessary will be useful if the stuck match lines are sensitive to the patterns. The further process flows assumes that there is no 0-pattern was applied. At the end of the process therefore the memory will be left with a 1-pattern.

[0024] Fig. 5 is a flowchart illustrating the weak pull down test S32 according to an embodiment of the invention. The objective of this test is to identify and eliminate from use in the CAM device 210 any defective (i.e., weak) pull down lines (i.e., conductive paths from the match line to ground). As can be seen from Fig. 3, this test is intended to be executed after the stuck match line test shown in Fig. 4 and, thus, assumes that each data storage location 110 in the CAM device 210 is set to 1-pattern and each mask M should also be set to unmasked (which is the state the cells 100 of the CAM device 210 were set to by the stuck match line test S31.)

[0025] The weak pull down test S32 begins at step S51, which, in cooperation with steps S55 and S56 sets up a for-loop iterating through steps S52-54, as described below. The index i is set to one during the first iteration and the loop terminates when i is incremented such that it exceeds the width of a word in the CAM device 210. At step S52, a search is performed using a search pattern. The search pattern is constructed so that each

bit in the search pattern compliment of the corresponding 1-pattern, except for bit-i, which is set to equal as corresponding pattern i. This pattern will be referred to as Walking-0 pattern. At step S53, the search results are examined to determine whether there is a match. If a match was detected, the method S32 has detected an error and the error is managed in step S54.

[0026] An error may be managed by repairing the matched i-th address bit if the CAM device 210 has sufficient redundant resources to affect a repair. Alternatively, the defective row may be disabled. Additionally, a record of the failure may be maintained by the tester 250. After the error has been managed in step S54, execution resumes at S52. The index i has not been incremented so that any steps taken in step S54 to manage the error is now tested to ensure that the error has been properly processed.

[0027] If no match was detected in step S53, the index i is incremented by one in step S55. If at S56 the index i is not greater than the width of the CAM words, then an iteration is completed and the process loops to step S52. If at S56 the index i is greater than the width of the CAM words, then the loop has completed all its iterations and execution continues at step S57.

[0028] At step S57, the CAM is written so that each CAM cell stores 0-pattern. The mask, value, previously set to one in step S42 if the device 210 is a ternary CAM, is left unchanged. As before, if the device 210 is a binary CAM, there will be no mask.

[0029] Steps S58, S62, and S63 form a for-loop much like steps S51, S55, and S56, as previously described. Execution begins with a first iteration at step S59, where a search is executed with a search expression set to have each bit compliment of 0-pattern except for bit-i, which is equal to the bit i. This pattern will be referred to as a Walking-1 pattern. In step S60, if a match is found there is an error and the error is managed by executing step S61. Step S61 is identical to step S54 (previously described). If no match is detected at 560, the for-loop continues with steps S62 and S63, which operate similarly to steps S55-

S56. The end of execution of the for-loop comprising steps S58, S62, and S63 ends the weak pull down test S32.

[0030] Figs. 6A and 6B are flowcharts illustrating a row-by-row test S33 according to an embodiment of the invention. The test S33 begins at step S600, which in combination with steps S604 and S605 form a for-loop iterating loop index j from one to height, the number of rows in the CAM device 210. Steps S601-S603 are the steps which are repated in the loop. In step S601, a search is performed using a search pattern which has 0-pattern values. In step S602, an inquiry as to whether a match was reported at row j is made. If no match was found, execution jumps to step S620. However, if there is a match, execution continues at step S603. Step S603 writes 1-pattern into all portions of row j. At step S604, the loop index j is incremented. At step S605, the value of the index j is check against the height parameter. If the index j has not yet exceeded the height parameter, execution resumes at step S601. Otherwise, execution continues at step S606.

[0031] Steps S606-S611 are similar to steps S600-S605. Step S606, in combination with steps S610 and S611 form a for-loop iterating loop index j from one to height. In step S607, a search is performed using a 1-pattern search pattern which has all one values. In step S608, an inquiry is made as to whether there is a match at row j. If not, execution continues at step S629. However, if there is a match, execution continues at step S609. Step S609 writes a 0-pattern into all portions of row j. At step S610, the loop index j is incremented. At step S611, the value of index j is checked against the height parameter. If index j has not yet exceeded the height parameter, execution resumes at step S607. Otherwise, the test S33 has completed.

[0032] When execution jumps to step S620 from step S602 (Fig. 6A) it is an indication that an error has been detected. Steps S620-S628 form a more detailed error analysis and handling routine. At step S620, a for-loop in cooperation with steps S626 and S627 is established, iterating loop counter k. (This loop is nested within the for-loop using loop counter j at steps S600, S604, and S605 of Fig. 6A.) At step S621, the test S33 writes

a pattern into row j which has zero at bit-k and one at every other bit position. If the device 210 is a ternary CAM, the mask M is set so that M=1 for bit-k and M=0 for all other bits in row j. At step S622, the test S33 performs a search using a pattern Walking-0_k At S623 the test S33 determines whether there is a match at row j. If so, counter k is incremented at step S626. At step S627 a check is made to see whether k is greater than width. If k is not greater than width, then execution continues at step S621. If k is greater than width, the test S33 failed to find an error we know which exists (S628). Execution therefore terminates.

[0033] If at step S623 there was no match at row j, the test S33 located the k-th bit in row j which is erroneous. At step S624 the test S33 decides whether to repair (e.g., remap) row-j, column-k, in the CAM device 210. If a repair is performed, execution continues at step S625, which writes a pattern of all zeros to row j. Execution then transfers to step S601. If row j is disabled, execution resumes at step S604 (Fig. 6A).

[0034] When execution jumps to step S629 from step S608 (Fig. 6A), it is an indication that an error has been detected. Steps S629-S637 form a more detailed error analysis and handling routine. At step S629, a for-loop in cooperation with steps S635 and S636 is established, iterating loop counter k. (This loop is nested within the for-loop using loop counter j at steps S606, S610 and S611 in Fig. 6A). At the S630, a pattern is written into row j which has one at bit-k and a zero at every other bit position. If the device 210 is a ternary CAM, the mask M is set so that M=1 for bit-k and M=0 for all other bits in row j. At step S631, a search using Walking-1 pattern is initiated. In step S632 it is determined whether there is a match at row j. If there is a match, k is incremented at step S635. The counter k is then checked to see if k is greater than width at step S636. If not, execution continues at step S630. If k is greater than width, then the test S33 has failed to find an error known to exist (S637). Execution therefore terminates.

[0035] If at step S632, there is no match at row j, the test S33 has located the k-th bit in row j which is erroneous. At step S633, it is decided whether to repair bit k in row j or

to disable row j. If it is decided to disable row j, execution resumes at step 610. If at S633 it is decided to repair bit k, the test S33 writes all ones to row j. Execution then transfers to step 607.

[0036] The above described processing can be used to test each cell 100 of the CAM device 210. During the error handling process, and more specifically, after an error has been detected but before the bit having the error is repaired, or the row having the error is remapped, a more detailed analysis of why the cell 100 has failed can be made using Table 1. Referring also to Fig. 1, the "Fault" column of Table 1 identifies transistor or line faults by name. In the case of transistor faults, the open or short fault condition is also identified. Each of these conditions can be tested by placing the indicated signals on the storage portion 110 (for D and D#), mask register M, and search data lines SD, and SD#. The results are determined by reading the state of the match line ML.

[0037] For example, referring to the first row under "Fault" (in Table 1), the test of the match line ML is performed by setting the mask M to zero and reading the associated match line. If the match line is zero, there is a fault with the match line ML. If the match line is set to one, then the match line ML is good (i.e., no error). Now referring to the fourth row (under column "Fault"), the test for "M1 open," if the indicated data, mask, and search data signals were asserted as listed in Table 1, and if the match line ML value is zero, Table 1 indicates that transistor M1 does not have an open fault. However, if the match line value was instead a one, Table 1 indicates that the transistor M1 has an open fault.

[0038] By using the conditions listed in Table 1, each faulty memory cell can be tested to identify some of the potential causes for failure. This data may help production engineers to fine tune the production process to avoid the identified errors.

Table 1: CAM Cell Fault Modeling							
Fault	D	D#	M	SD	SD#	ML	
						Good	Fail
ML	X	X	0	X	X	1	0
	0	1	X	1	0		
	1	0	X	0	1		
	1	0	1	1	0	0	1
	0	1	1	0	1		
SD	1	0	1	1	0	0	1
	1	0	1	0	1	1	0
SD#	0	1	1	0	1	0	1
	0	1	1	1	0	1	0
M1 open	1	0	1	1	0	0	1
M1 short	0	1	1	1	0	1	0
M1# open	0	1	1	0	1	0	1
M1# short	1	0	1	0	1	1	0
M2 open	0	1	1	0	1	0	1
M2 short	1	0	1	1	0	1	0
M2# open	0	1	1	0	1	0	1
M2# short	0	1	1	1	0	1	0
M3 open	1	0	1	1	0	0	1
	0	1	1	0	1		
M3 short	1	0	0	1	0	1	0
	0	1	0	0	1		

[0039] Thus, the present invention describes a methodology for testing CAM memory cells in a CAM device. A match line test is first performed to identify and manage stuck match line. Then a weak pull-down test is performed to identify and manage faulty pull-down lines. Once the mach lines and pull down lines are assumed to be functional, a row-by-row test is conducted. When a cell is identified as having failed, Table 1, above provides

a mapping of signal states whereby reading the match line associated with the CAM cell can be used to identify the type of error.

[0040] While the invention has been described in detail in connection with the exemplary embodiment, it should be understood that the invention is not limited to the above disclosed embodiment. Rather, the invention can be modified to incorporate any number of variations, alternations, substitutions, or equivalent arrangements not heretofore described, but which are commensurate with the spirit and scope of the invention. Accordingly, the invention is not limited by the foregoing description or drawings, but is only limited by the scope of the appended claims.